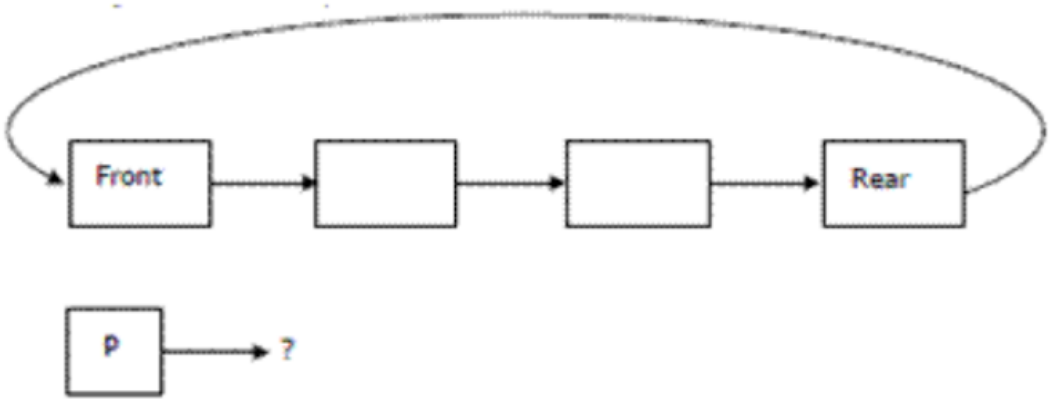
Data Structures Mid Sem

Your email will be recorded when you submit this form.

Not **lit2020036@iiitl.ac.in**? Switch account

A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations enQueue and deQueue can be performed in constant time?

2 points



- (●) a) rear node
- ( ) b) front node
- ( ) c) not possible with a single pointer
- ( ) d) node next to the front

Clear selection

An array A consists of n integers in locations A[0], A[1] ....A[n-1]. It is required to shift the elements of the array cyclically to the left by k places, where 1 <= k <= (n-1). An incomplete algorithm for doing this in linear time, without using another array is given below. Complete the algorithm by filling in the blanks. Assume alt the variables are suitably declared. 2 points

```
min = n; i = 0;
while (___________) {
     temp = A[i]; j = i;
     while (________) {
     A[j] = ________
     j= (j + k) mod n ;
     If ( j< min ) then
         min = j;
}
A[(n + i – k) mod n] = _________
i = __________
```

- (A) i > min; j!= (n+i)mod n; A[j + k]; temp; i + 1 ;
- (B) i < min; j!= (n+i)mod n; A[j + k]; temp; i + 1;
- (C) i < min; j!= (n+i-k)mod n; A[(j + k)mod n]; temp; i + 1;
- (D) i > min; j!= (n+i+k)mod n; A[(j + k)]; temp; i + 1;

Clear selection

Name

Prabhav Agrawal

A priority queue Q is used to implement a stack that stores characters. PUSH (C) is implemented INSERT (Q, C, K) where K is an appropriate integer key chosen by the implementation. POP is implemented as DELETEMIN(Q). For a sequence of operations, the keys chosen are in 2 points

- ○ non-increasing order
- ○ non-decreasing order
- ○ strictly increasing order
- ◉ strictly decreasing order

Clear selection

Consider a two dimensional array A[20][10]. Assume 4 words per memory cell, the base address of array A is 100, elements are stored in row-major order and first element is A[0][0]. What is the address of A[11][5] ? 2 points

- ◉ (A) 560
- ○ (B) 460
- ○ (C) 570
- ○ (D) 575

Clear selection

Following is C like pseudo code of a function that takes a Queue as an argument, and uses a stack S to do processing. 3 points

```
void fun(Queue *Q)
{
    Stack S;  // Say it creates an empty stack S

    // Run while Q is not empty
    while (!isEmpty(Q))
    {
        // deQueue an item from Q and push the dequeued item to S
        push(&S, deQueue(Q));
    }

    // Run while Stack S is not empty
    while (!isEmpty(&S))
    {
      // Pop an item from S and enqueue the poppped item to Q
      enQueue(Q, pop(&S));
    }
}
```

- (A) Removes the last from Q
- (B) Keeps the Q same as it was before the call
- (C) Makes Q empty
- (D) Reverses the Q

Clear selection

Consider an array consisting of −ve and +ve numbers. What would be the worst case time complexity of an algorithm to segregate the numbers having same sign altogether i.e all +ve on one side and then all -ve on the other ? 2 points

- [x] (A) O(N)
- [ ] (B) O(N Log N)
- [ ] (C) O(N * N)
- [ ] (D) O(N Log Log N)

Clear selection

Consider the following C functions in which size is the number of elements in the array E: 5 points

```
int MyX(int *E, unsigned int size){
 int Y = 0;
 int Z;
 int i,j,k;
 for(i = 0; i < size; i++)
       Y = Y + E[i];
 for(i = 0; i < size; i++)
   for(j = i; j < size; j++){
     Z = 0;
     for(k = i; k <= j; k++)
        Z = Z + E[k];
     if(Z > Y)
        Y = X;
   }
   return Y;
}
```

The value returned by the function MyX is the

- (●) Maximum possible sum of elements in any sub-array of array E
- ( ) Maximum elements in any sub-array of array E
- ( ) Sum of the maximum elements in all possible sub-array of array E
- ( ) The sum of all elements of array E

Clear selection

The postfix expression for the infix expression A + B * (C + D) / F + D * E is: 4 points

ABCD+*F/+DE*+

The subset-sum problem is defined as follows. Given a set of n positive integers, S = {a1 ,a2 ,a3 ,...,an} and positive integer W, is there a subset of S whose elements sum to W? A dynamic program for solving this problem uses a 2-dimensional Boolean array X, with n rows and W+1 columns. X[i, j],1 <= i <= n, 0 <= j <= W, is TRUE if and only if there is a subset of {a1 ,a2 ,...,ai} whose elements sum to j. Which of the following is valid for 2 <= i <= n and ai <= j <= W? 2 points

- (A) X[i, j] = X[i − 1, j] V X[i, j -ai]
- (B) X[i, j] = X[i − 1, j] V X[i − 1, j − ai]
- (C) X[i, j] = X[i − 1, j] V X[i, j − ai]
- (D) X[i, j] = X[i − 1, j] V X[i -1, j − ai]

Clear selection

Consider the following statements:(i) First-in-first out types of computations are efficiently supported by STACKS.(ii) Implementing LISTS on linked lists is more efficient than implementing LISTS on an array for almost all the basic LIST operations.(iii) Implementing QUEUES on a circular array is more efficient than implementing QUEUES on a linear array with two indices.(iv) Last-in-first-out type of computations are efficiently supported by QUEUES. 2 points

- i and iii are true
- i and ii are true
- iii and iv are true
- ii and iv are true

Clear selection

A function f defined on stacks of integers satisfies the following properties. f(∅) = 0 and f (push (S, i)) = max (f(S), 0) + i for all stacks S and integers i. If a stack S contains the integers 2, -3, 2, -1, 2 in order from bottom to top, what is f(S)? 2 points

- ○ 6
- ○ 4
- ◉ 3
- ○ 2

Clear selection

The time complexity of the following C function is (assume n > 0) 2 points

- ○ a) 0(n)
- ○ b) 0(nlogn)
- ○ c) 0(n^2)
- ◉ d) 0(2^n)

Clear selection

Roll No.

LIT2020036

Submit